A Key Sharing Fuzzy Vault Scheme

Lin You[1], Mengsheng Fan[1], Jie Lu[2], Shengguo Wang[2], and Fenghai Li[3]

[1] College of Comm. Engr., Hangzhou Dianzi Univ., Hangzhou 310018, China
[2] Zhejiang Wellcom Technology Co., Ltd, Hangzhou 310012, China
[3] The Key Lab. of Information Assurance Technology, Beijing 10072, China

Abstract. Based on the classical fuzzy vault and the Diffie-Hellman key exchange scheme, a key sharing fuzzy vault scheme is proposed. In this fuzzy vault scheme, the two users cooperatively build their shared fuzzy vault with a shared key hidden in it using their own biometric features, and they can respectively use their biometric features to unlock the fuzzy vault to get their shared key without running the risk of disclosure of their biometric features later. The security of our scheme is based on the security of the classical fuzzy vault scheme and and the discrete logarithm problem in a given finite group.

Keywords: Fuzzy Vault, Diffie-Hellman key exchange, Finite group, Biometrics, Polynomial interpolation.

1 Introduction

In a cryptosystem, one of the most important procedure is to securely store the secret key. Generally, the secret key is stored in the user's computer, a smart card or other storage medias by using a password for accessing, but it will run the risks that the storage medias be lost or stolen, or the password will suffer from the exhaustive search attack. A better way is to use the user's biometric features as the access control measure, while the user's biometric feature or secret key may also be disclosed if his biometric template and key are separately stored. Therefore, to ensure their safety simultaneously, the user's biometric feature and secret key should be completely blended into one set or a data. A classical solution is the fuzzy vault proposed by Juels and Sudan in 2002 [1]. In their fuzzy vault scheme, they used the user's unique set to blend his secret into a vault based on Reed-Solomon codes, and the user can recover his secret by providing a set that overlaps largely with the original set. Even if an attacker can get the vault he cannot obtain the the user's secret or the information about the set.

Diffie-Hellman key exchange scheme is a key cryptographic protocol, but how to safely store the shared key between the users is also a thorny problem. In order to produce a shared key between two parties and protect it from being illegally exposed, based on the ideals of the original fuzzy vault and the Diffie-Hellman key exchange scheme, a fuzzy vault scheme for the secret key exchange is proposed in this work. The security of this fuzzy vault scheme is based on both a polynomial reconstruction problem and a discrete logarithm problem.

T.W. Chim and T.H. Yuen (Eds.): ICICS 2012, LNCS 7618, pp. 453–460, 2012.

In the following Section 2, the classical fuzzy vault scheme is introduced. Then, our key sharing fuzzy vault scheme is proposed in Section 3 and its security analysis is given in Section 4. Finally, some concluding remarks are presented in Section 5.

2 The Classical Fuzzy Vault Scheme

The classical fuzzy vault scheme was invented by Juels and Sudan in 2002 and was revised in 2006 [2]. Essentially, the fuzzy vault is a scheme for the secure protection of one's secret (value or key) by the use of his some private message set which generally comes from his unique biometrics. A fuzzy vault is composed of two algorithms, one is called the locking algorithm, and the other is called the unlocking algorithm, as the following Fig. 1 and Fig. 2 shown, respectively.

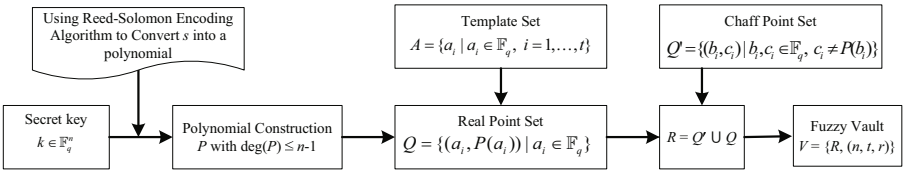

Fig. 1. Juels & Sudan's Fuzzy Vault Scheme–Locking Algorithm

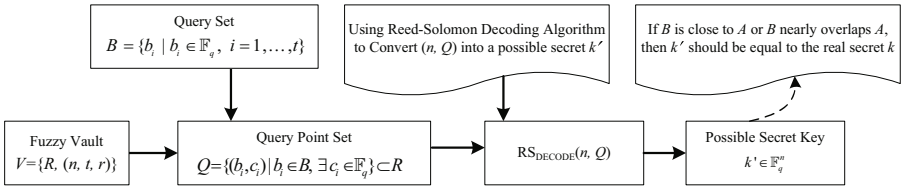

Fig. 2. Juels & Sudan's Fuzzy Vault Scheme–Locking Algorithm

A fuzzy vault scheme includes two public parameters, one is a finite field $\mathbb{F}_q$ with q a power of a prime, and the other is a Reed-Solomon decoding algorithm (denoted as $\text{RS}_{\text{DECODE}}$ for short). The most practical choice for $\text{RS}_{\text{DECODE}}$ is the Reed-Solomon decoding algorithm based on Newton's interpolation [3] or the Lagrange interpolation polynomial. The following two algorithms for the fuzzy vault scheme comes originally from the revised work of Juels and Sudan [2] except for some minor changes. The security of this scheme is based on a polynomial reconstruction problem.

2.1 Locking Algorithm

INPUT: Parameters n, t, and r such that $n \leq t \leq r \leq q$, a pre-selected secret key $k \in \mathbb{F}_q^n$, a set $A = \{a_i\}_{i=1}^t$ with $a_i \in \mathbb{F}_q$ being distinct.

OUTPUT: A fuzzy vault $V = \{R, (n, r, q)\}$ with R being a set of points $\{(x_i, y_i)\}_{i=1}^r$ such that $x_i, y_i \in \mathbb{F}_q$ and all x_i being distinct.

1. X, R, $V \leftarrow \emptyset$;
2. $P \leftarrow k$, that is, k is block-encoded into the coefficients of a polynomials of degree n in $\mathbb{F}_q$;
3. For $i = 1$ to t do
   - $(x_i, y_i) \leftarrow (a_i, P(a_i))$;
   - $X \leftarrow X \bigcup \{x_i\}$;
   - $R \leftarrow R \bigcup \{(x_i, y_i)\}$;

   for $i = t + 1$ to r do
   - $x_i \in_U \mathbb{F}_q \backslash X$;
   - $X \leftarrow X \bigcup \{x_i\}$;
   - $y_i \in_U \mathbb{F}_q \backslash \{P(x_i)\}$;
   - $R \leftarrow R \bigcup \{(x_i, y_i)\}$.
4. Output R or $V = \{R, (n, r, q)\}$.

In order not to leak information about the order in which the x_i are chosen, the set R should be output in a pre-determined order, e.g., the points in R may be arranged in order of ascending x-coordinates, or else in a random order. Note that the chaff points in the locking algorithm should be selected so as to intersect neither the set A nor the polynomial P is for the security consideration. Generally, the set V combining the set R and the triple vector (n, r, q) is called a fuzzy vault.

2.2 Unlocking Algorithm

INPUT: A fuzzy vault V comprising a parameter pair (n, r, q) such that $n \leq r \ll q$ and a set R of r points with their two coordinations in $\mathbb{F}_q$. A query set $B = \{b_i\}_{i=1}^{t}$ with $b_i \in \mathbb{F}_q$.
OUTPUT: An element $k' \in \mathbb{F}_q^n \bigcup \{\text{'null'}\}$.

1. $Q \leftarrow \emptyset$;
2. For $i = 1$ to t do
   - If there exists some $y_i \in \mathbb{F}_q$ such that $(b_i, y_i) \in R$, set $Q \leftarrow Q \bigcup \{(b_i, y_i)\}$;
   - Set $k' \leftarrow$ '*null*' if Q has less than n points;
   - Otherwise, set $k' \leftarrow \mathrm{RS}_{\mathrm{DECODE}}(n, Q)$;
3. Output k'.

Suppose that the fuzzy vault V is created by Alice and Bob tries to unlock V to recover the secret key k. Bob has to use his set B to determine the codeword that encodes the secret key k to get a possible secret key k'. Since the set A specifies the x-coordinates of "correct" points that lie on the polynomial P. Thus, if B is close to A, then B will identify a large majority of these "correct" points. Any divergence between B and A will introduce a certain amount of error. However, this noise may be removed by means of a Reed-Solomon decoding algorithm provided that there is sufficient overlap.

The most convenient and unique features to the user is his biometric feature set, such as the fingerprint features, iris features, retinal features and etc. In 2005, Uludag and *et al.* [4] proposed a fingerprint-based fuzzy vault. One can also use our other biometric features to construct fuzzy vault schemes.

3 A Key Sharing Fuzzy Vault Scheme

The most popular and classical key sharing scheme is the Diffie-Hellman key exchange scheme [5] which is a specific method for sharing a secret key between two parties, and it is one of the earliest practical examples of secret key exchange or secret key scheme implemented within the field of cryptography. The Diffie-Hellman key exchange method allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel. This established shared (secret) key can later be used in any symmetric key algorithm.

In practical applications, the multiplicative group G is generally chosen to be a multiplicative group $\mathbb{F}_q^*$ with q a power of a prime. To increase its security strength, we can set up the key sharing scheme on an elliptic curve rational point group or a hyperelliptic curve Jacobian group since the discrete logarithm problem is much harder than the discrete logarithm problem in the multiplicative group of a Galois field.

In this section, we will put out a novel fuzzy vault scheme for secret key sharing scheme based on the classical fuzzy vault and a multiplicative group, here we denote this scheme as KSFV scheme.

We suppose that Alice and Bob want to establish a shared secret key for their future cryptographic applications by using their biometric features, such as their fingerprint features, then they agree on a finite multiplicative group $G = \mathbb{F}_q^*$ with q a power of a large prime and a cyclic subgroup $< g >$ of G with g an element of some large prime order p. Here, G, q, g and p are assumed to be public parameters.

3.1 Locking Algorithm

INPUT: A finite multiplicative group $G = \mathbb{F}_q^*$ with q a prime power and one of its cyclic subgroup $H =< g >$ of large prime order p; Positive integers n, s, t, r_A and r_B satisfying $n \leq min\{s, t\} \leq s + t \leq r_A, r_B \ll p$; All these parameters are made public.

OUTPUT: $V = \{R_{AB}, (p, g, n)\}$, where R_{AB} is a set composed of much more than n points with their coordinations in $\mathbb{F}_q^*$.

1. X, $\bar{X}$, R, R_A, R_B, $V \leftarrow \emptyset$;
2. Alice and Bob extract their private biometric features $A = \{a_i\}_{i=1}^s$ and $B = \{b_j\}_{j=1}^t$, respectively;
3. Convert a_i and b_j $(i = 1, \ldots, s, j = 1, \ldots, t)$ into the elements in $\{2, \ldots, p-1\}$. For convenience, they are still respectively represented as a_i and b_j which are supposed to be different from each others, and the corresponding sets are still respectively denoted as A and B.
4. Alice randomly selects a select key $a \in \{2, \ldots, p-1\}$, computes g^a and sends it to Bob;
5. For $i = 1, \ldots, s$, Alice compute $g^{a_i} (\triangleq \alpha_i)$ and sends the results to Bob;

6. Bob randomly selects a select key $b \in \{2, \ldots, p-1\}$, computes g^b and sends it to Alice;
7. For $j = 1, \ldots, t$, Bob computes $g^{b_j} (\triangleq \beta_j)$ and sends the results to Alice;
8. Alice and Bob compute $(g^b)^a$ and $(g^a)^b$, respectively;
9. For each fixed $j \in \{1, \ldots, t\}$, Alice computes $(\beta_j)^{a_i}$ and set it to $\alpha_{j,i}$ for $i = 1, \ldots, s$;
10. For each fixed $i \in \{1, \ldots, s\}$, Bob computes $(\alpha_i)^{b_j}$ and set it to $\beta_{i,j}$ for $j = 1, \ldots, t$;
11. For $i = 1, \ldots, s$ and $j = 1, \ldots, t$, set $\gamma_{i,j} = \alpha_{i,j}$ (Obviously, we have $\alpha_{j,i} = g^{a_i b_j} = \beta_{i,j}$);
12. $k \leftarrow g^{ab}$ (Since $(g^b)^a = g^{ba} = g^{ab} = (g^a)^b$, k can be regarded as Alice and Bob's shared key);
13. Alice and Bob, respectively, set $P(x) \leftarrow k$. That is, k is block-encoded into the coefficients of a polynomial of degree n in $\mathbb{F}_p[x]$;
14. Alice does the following steps:
    (a) For $j = 1$ to t, $i = 1$ to s do
        - $(x_{i+j}, y_{i+j}) \leftarrow (\gamma_{i,j}, P(\gamma_{i,j}))$;
        - $X \leftarrow X \bigcup \{x_{i+j}\}$;
        - $R \leftarrow R \bigcup \{(x_{i+j}, y_{i+j})\}$;
    (b) For $l = s + t + 1$ to r_A do
        - $x_l \in_U < g > \backslash X$;
        - $\bar{X} \leftarrow \bar{X} \bigcup \{x_l\}$;
        - $y_l \in_U < g > \backslash \{P(x_l)\}$;
        - $R_A \leftarrow R \bigcup \{(x_l, y_l)\}$.
    (c) Alice sends R_A to Bob.
15. In the meantime, Bob does the similar steps to generate R_B with the same real point set R and $r_B - (s + t)$ chaff pints. R_B is sent to Alice;
16. Set $R_{AB} = R_A \bigcup (R_B \backslash R)$. (Note that $R_{AB} = (R_A \bigcup R_B) \backslash R = R_B \bigcup (R_A \backslash R)$);
17. Output $V = \{R_{AB}, (p, g, n)\}$.

The output V is regarded as the key sharing fuzzy vault owned by both Alice and Bob. If one of them wants to restore the shared key k, he/she can independently use his/her own biometrics to restore the possible shared sky k' by the following "Unlocking Algorithm".

3.2 Unlocking Algorithm

INPUT: A finite multiplicative group $G = \mathbb{F}_q^*$ and one of its cyclic subgroup $< g >$ of large prime order p; Alice and Bob's biometric sets $A' = \{a_i'\}_{i=1}^{s'}$ and $B' = \{b_j'\}_{j=1}^{t'}$ with a_i', $b_j' \in \{2, \ldots, p-1\}$, respectively; A set $V = \{R_{AB}, (p, g, n)\}$ satisfying that $n \leq s', t' < s' + t' \ll p$, and the all points in R_{AB} are in $\mathbb{F}_p^* \times \mathbb{F}_p^*$.

OUTPUT: An element $k' \in \mathbb{F}_p^* \bigcup \{\text{'}null\text{'}\}$.

1. $Q \leftarrow \emptyset$;
2. If Alice and Bob want to recover the shared key k, they do the following:

(a) For $i = 1$ to s', Alice computes $g^{a'_i}$ ($\triangleq \alpha'_i$) and send α'_i to Bob;
(b) For $j = 1$ to t', Bob computes $g^{b'_j}$ ($\triangleq \beta'_j$) and send β'_j to Alice;
(c) For each fixed $j \in \{1, \ldots, t'\}$, Alice computes $(\beta'_j)^{a'_i}$ and set it to $\beta'_{i,j}$ for $i = 1, \ldots, s'$;
(d) For each fixed $i \in \{1, \ldots, s'\}$, Bob computes $(\alpha'_i)^{b'_j}$ and set it to $\alpha'_{i,j}$ for $j = 1, \ldots, t'$;
(e) Alice does the following:
  i. If there exists some $y \in \mathbb{F}_q^*$ such that $(\alpha'_{i,j}, y) \in R_{AB}$, do
    – $(x_{i+j}, y_{i+j}) \leftarrow (\alpha'_{i,j}, y)$;
    – $Q \leftarrow Q \bigcup \{(x_{i+j}, y_{i+j})\}$.
  ii. $k' \leftarrow \mathrm{RS}_{\mathrm{DECODE}}(n, Q)$ (For example, one can apply Newton's interpolation polynomial or Lagrange interpolation polynomial to get a possible key k' if Q has no less than n points.);
  iii. $k' \leftarrow$ '*null*' if Q has less than n points.
(f) $k' \leftarrow \mathrm{RS}_{\mathrm{DECODE}}(n, Q)$ or '*null*'.

3. Similarly, Bob can do the similar steps as Alice does to recover the possible shared key k'.
4. Output k'.

The locking algorithm and unlocking algorithm can be described as the following Fig.3 and Fig.4, respectively. Here, the used biometrics are supposed to be the users' fingerprints and Lagrange interpolation polynomial is used for the Reed-Solomon decoding algorithm.

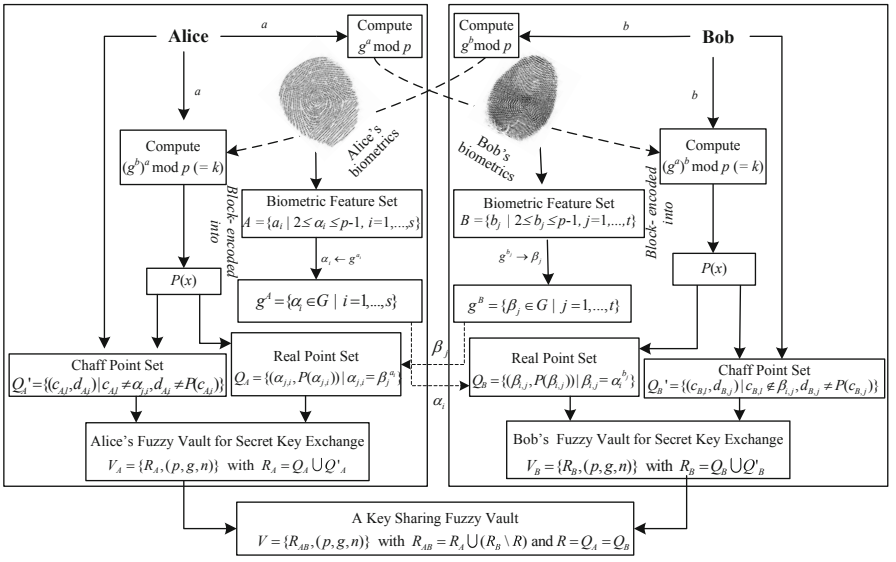

Fig. 3. KSFV-Locking Algorithm

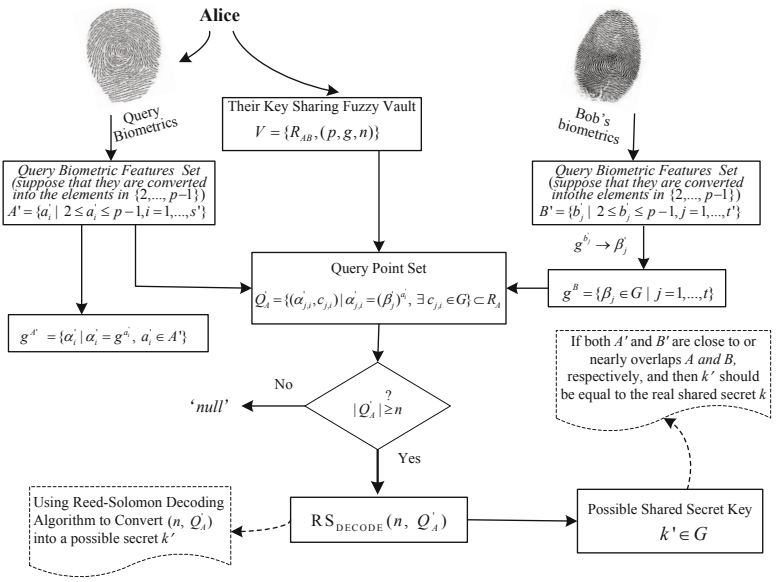

Fig. 4. SKFV-Unlocking Algorithm (for Alice)

If Alice and Bob can provide their biometric sets A' and B' that are respectively close to or sufficiently overlap A and B, that is, if both of their biometric sets A' and B' contain no less than n "correct" biometric features, then they will recover their real shared key k successfully. Otherwise, they will fail to recover a right shared key.

According to Guruswami and Sudan's polynomial reconstruction algorithm [6], if the query set Q contains at least $\min\{\sqrt{ns'}, \sqrt{nt'}\}$ "correct" or real points, then there exists a polynomial time algorithm to reconstruct the correct polynomial $P(x)$, and it follows that the real shared key k can be recovered successfully.

4 Security Analysis

From the construction of our KSFV scheme, one can see that its security is based on both the security of the classical fuzzy vault scheme and the discrete logarithm problem (DLP).

Firstly, the security of our KSFV construction depends on the number of chaff points $r_A + r_B - 2(s+t)$ in the target set R_{AB} of the total points $r_A + r_B - s - t$. The greater the number of such points, the more noise there is to conceal the real polynomial $P(x)$ from an attacker. As many chaff points are added to R_{AB}, there will be a set of many spurious polynomials that look like $P(x)$. In the absence of additional favorable information,the probability, that an attacker can obtain the real polynomial is $\binom{s+t}{n+1} / \binom{r_A+r_B-s-t}{n+1}$ or $\prod_{i=0}^{n-1} \frac{s+t-i}{r_A+r_B-s-t-i}$. Since both r_A and r_B are taken much larger than n and $s+t$, the probability is approximate

to $(\frac{s+t}{r_A+r_B})^n$ which becomes much smaller as r_A or r_B gets much larger. That is, the security is proportional to the number of spurious polynomials.

For some more detail security analysis on the classic fuzzy vault, one can refer to Juels and Sudan's work (the section 4 in [2]).

Secondly, the shared key k is produced based on Diffie-Hellman key exchange scheme on a cyclic group H of a large prime p, an attacker can only get k if he could solve the discrete logarithm problem on H. In addition, since the two users' biometric features are not directly transferred to each other or stored in our novel fuzzy vault, but they are hiddenly transferred to the other party by the exponent calculations with the user's biometric numbers as the exponents. Hence, to access to the users' biometrics features is equivalent to solve the discrete logarithm problems on H.

5 Conclusion

Based on fuzzy vault scheme and Diffie-Hellman key exchange scheme, a key sharing fuzzy vault scheme for secure key sharing scheme is proposed in this work. The security of this fuzzy vault scheme is based on both the security of the classical fuzzy vault scheme and the discrete logarithm problem. This key sharing fuzzy vault scheme is just a detailed model but it will be simulated for fingerprints in our future work. In addition, similar to our method, a key sharing fuzzy vault scheme for the multiparty secret sharing protocol can also be set up.

Acknowledgments. This work is partially supported by the Research Projects of Zhejiang Natural Science Foundations (No.R10900138) and The Key Laboratory of Information Assurance Technology (KJ-11-05).

References

1. Juels, A., Sudan, M.: A fuzzy vault scheme. In: IEEE International Symposium on Information Theory (ISIT), p. 408. IEEE Press, Lausanne (2002)
2. Juels, A., Sudan, M.: A fuzzy vault scheme. Designs, Codes, and Cryptography 38(2), 237–257 (2006)
3. Sorger, U.K.: A New Reed-Solomon Code Decoding Algorithm Based on Newton's Interpolation. IEEE Transactions on Information Theory 39(2), 358–365 (1993)
4. Uludag, U., Pankanti, S., Jain, A.K.: Fuzzy Vault for Fingerprints. In: Kanade, T., Jain, A., Ratha, N.K. (eds.) AVBPA 2005. LNCS, vol. 3546, pp. 310–319. Springer, Heidelberg (2005)
5. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Transactions on Information Theory 22(6), 644–654 (1976)
6. Guruswami, V., Sudan, M.: Improved decoding of Reed-Solomon and Algebraic-Geometric codes. IEEE Transactions on Information Theory 45(6), 1757–1767 (1999)